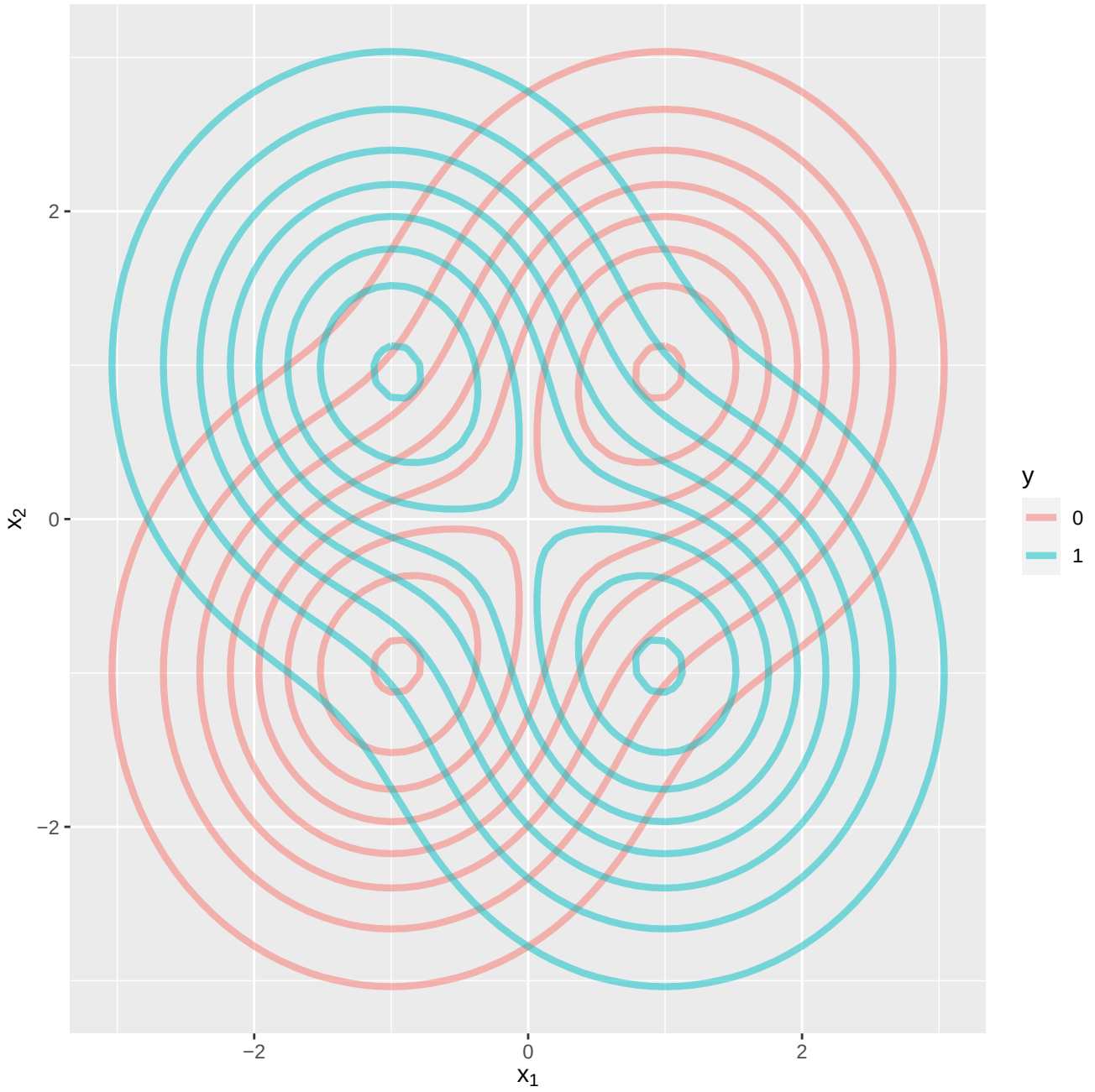
Solution 1: Filter problems



(a)

(b) Let g_μ be the density associated to $\mathcal{N}(\mu, 1)$. Note that, in general, for $\mathbf{x} \in \mathbb{R}^d \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ each component $x_i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_{ii})$. Also for finite mixtures, it holds that the marginal density of a mixture is the sum of the associated marginal densities since $\int \cdots \int \sum_{j=1}^{m} p_j(\boldsymbol{x}) dx_1 \cdots dx_{k-1} dx_{k+1} \cdots dx_d =$

$\sum_{j=1}^{m} \int \cdots \int p_j(\boldsymbol{x}) dx_1 \cdots dx_{k-1} dx_{k+1} \cdots dx_d$ where $p_1 \ldots p_m$ are densities. With this, it follows that

$$\begin{aligned}
\mathbb{P}(Y = 1 | x_i = \widetilde{x}_i) &= \frac{p(x_i = \widetilde{x}_i | Y = 1)\mathbb{P}(Y = 1)}{p(x_i = \widetilde{x}_i | Y = 1)\mathbb{P}(Y = 1) + p(x_i = \widetilde{x}_i | Y = 0)\mathbb{P}(Y = 0)} \\
&\overset{\mathbb{P}(Y=1)=\mathbb{P}(Y=0)}{=} \frac{p(x_i = \widetilde{x}_i | Y = 1)}{p(x_i = \widetilde{x}_i | Y = 1) + p(x_i = \widetilde{x}_i | Y = 0)} \\
&= \frac{0.5(g_{-1}(\widetilde{x}_i) + g_1(\widetilde{x}_i))}{0.5(g_{-1}(\widetilde{x}_i) + g_1(\widetilde{x}_i)) + 0.5(g_{-1}(\widetilde{x}_i) + g_1(\widetilde{x}_i))} = 0.5.
\end{aligned}$$

(c)

$$\begin{aligned}
\mathbb{P}(Y = 1 | x_1 = 1, x_2 = 1) &= \frac{p(\boldsymbol{x} = (1,1)^\top | Y = 1)\mathbb{P}(Y = 1)}{p(\boldsymbol{x} = (1,1)^\top | Y = 1)\mathbb{P}(Y = 1) + p(\boldsymbol{x} = (1,1)^\top | Y = 0)\mathbb{P}(Y = 0)} \\
&= \frac{1}{1 + \frac{p(\boldsymbol{x}=(1,1)^\top | Y=0)}{p(\boldsymbol{x}=(1,1)^\top | Y=1)}} \\
&= \frac{1}{1 + \frac{\exp(0) + \exp(-0.5(-2,-2)^\top(-2,-2))}{2\exp(-0.5(0,-2)^\top(0,-2))}} \\
&= \frac{1}{1 + \frac{\exp(0)+\exp(-4)}{2\exp(-2)}} \approx 0.21.
\end{aligned}$$

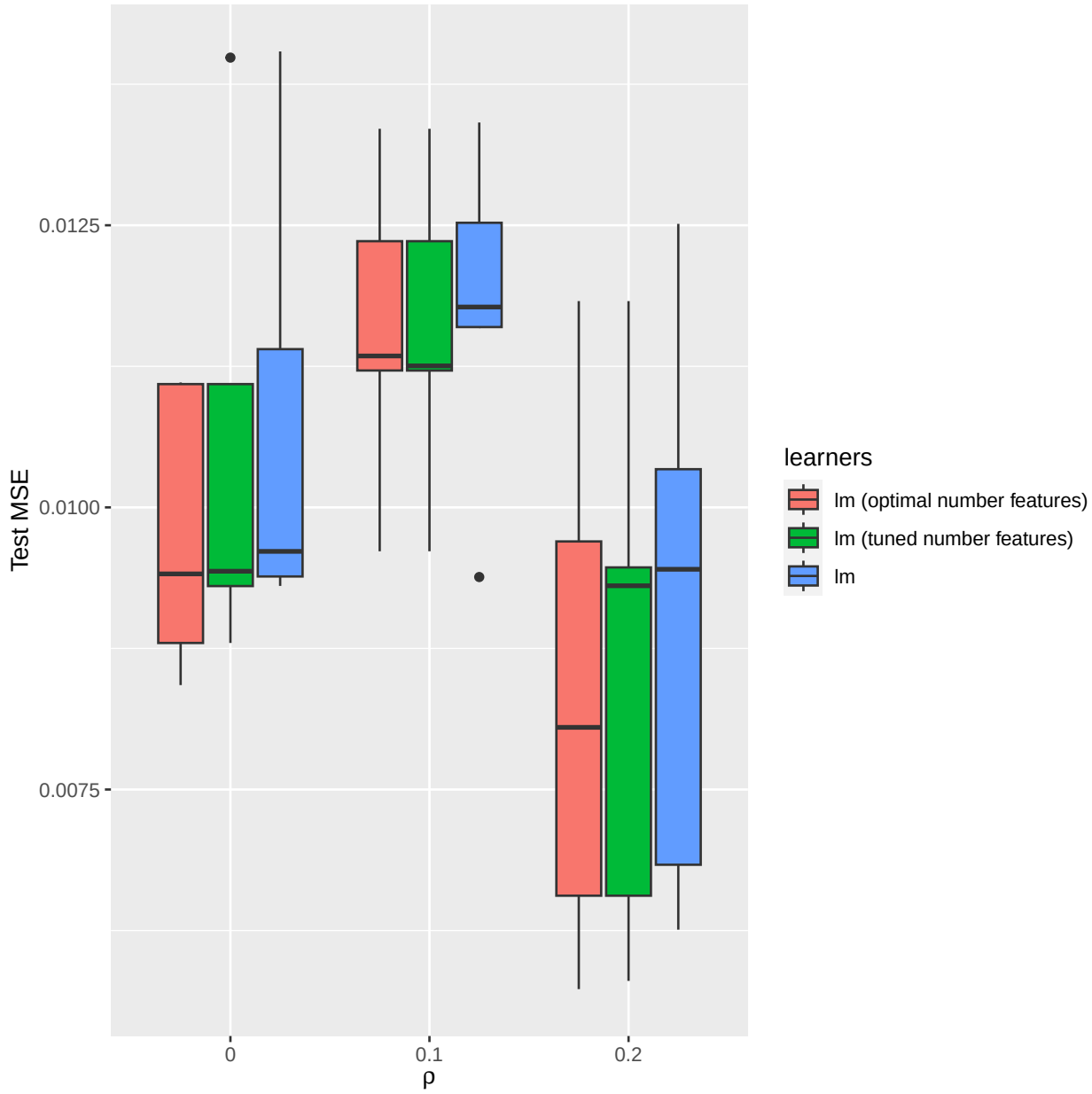
(d) It holds by b) that x_1 and x_2 are pairwise independent from Y since $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 1 | x_i = \widetilde{x}_i) = 0.5$. Hence the mutual information will be 0 for both features. So any other feature would be preferred over them, although Y is clearly jointly dependent on x_1, x_2, as shown in c).

Solution 2: Filter simulation

(a)

```
lgr::get_logger("mlr3")$set_threshold("warn")
lgr::get_logger("bbotk")$set_threshold("warn")

library(mlr3)
library(mlr3learners)
library(mlr3pipelines)
library(mlr3filters)
library(mlr3fselect)
library(ggplot2)
library(mlr3tuning)
library(mvtnorm)

set.seed(123)

# Define setup
p <- 10
frac <- 0.4
n <- 200
sigma_noise <- 0.1
num_benchmarks <- 5
# create sparse ground truth
beta <- c(rep(1, p * frac), rep(0, p * (1 - frac)))

resamplings <- rsmp("holdout")

# Define learners
po_flt <- po("filter", filter = flt("correlation"), filter.nfeat = p * frac)
graph_notune <- po_flt %>>% po("learner", lrn("regr.lm"))
```

```
po_filter <- po("filter", filter = flt("correlation"),
                  filter.nfeat = to_tune(1, p - 1))

graph <- as_learner(po_filter %>>% po("learner", lrn("regr.lm")))

graph_tune <- auto_tuner(tnr("grid_search"),
    graph,
    rsmp("holdout"),
    msr("regr.mse"),
    term_evals = p - 1
  )
learners <- list(lrn("regr.lm"), graph_notune, graph_tune)

# repeat benchmark
aggrs <- NULL
for (i in seq(1, num_benchmarks)){
  tasks <- list()
  # create tasks with varying correlation rho betweeen the features
  for (rho in seq(0, 0.2, 0.1)) {
    sigma <- matrix(rho, p, p)
    diag(sigma) <- 1.0

    x <- rmvnorm(n, sigma = sigma)
    y <- x %*% beta + sigma_noise * rnorm(n)
    tasks <- append(tasks, TaskRegr$new(as.character(rho),
      data.frame(x = x, y = y), "y"))
  }

  # do benchmark
  design <- benchmark_grid(tasks, learners, resamplings)
  bmr <- benchmark(design)

  # merge results
  aggr <- bmr$aggregate()
  aggrs <- rbind(aggrs, aggr[, c("task_id", "regr.mse",
    "learner_id")])
}

ggplot(aggrs) +
  geom_boxplot(aes(x = task_id, y = regr.mse,
    fill = learner_id)) + #ylim(c(0, 0.02)) +
    ylab("Test MSE") + xlab(expression(rho)) +
    scale_fill_discrete(labels = c(
      "lm (optimal number features)",
      "lm (tuned number features)",
      "lm")) +
      guides(fill = guide_legend(title = "learners"))
```



(b) The simulation study suggests that in this sparse scenario feature selection results in better performance (even if the true number of sparse features is unknown). This effect seems to be independent of the correlation between the features.

Solution 3: Wrappers

(a) Iterations:

(i) {B} since $\text{BIC}_{\{B\}} < \text{BIC}_X \quad \forall X \in \{\{A\}, \{C\}, \{D\}\}$,
(ii) {B, D} since $\text{BIC}_{\{B,D\}} < \text{BIC}_X \quad \forall X \in \{\{A, B\}, \{B, C\}\}$,
(iii) {B, C, D} since $\text{BIC}_{\{B,C,D\}} < \text{BIC}_{\{A,B,D\}}$,
(iv) {B, C, D} and terminate since $\text{BIC}_{\{B,C,D\}} < \text{BIC}_{\{A,B,C,D\}}$.

(b) Iterations:

(i) Start with all features {A, B, C, D}
(ii) {B, C, D} since $\text{BIC}_{\{B,C,D\}} < \text{BIC}_X \quad \forall X \in \{\{A, B, C\}, \{A, C, D\}\}$,

(iii) {B, C, D} and terminate since $\text{BIC}_{\{B,C,D\}} < \text{BIC}_X \quad \forall X \in \{\{B,C\},\{B,D\},\{C,D\}\}$,